Silk: A Simulation Study of Regulating Open Normative Multiagent Systems

Mehdi Mashayekhi, Hongying Du, George F. List, Munindar P. Singh
North Carolina State University, Raleigh, NC 27695, USA
{mmashay2, hdu2, gflist, singh}@ncsu.edu

Abstract

In a multiagent system, a (social) norm describes what the agents may expect from each other. Norms promote autonomy (an agent need not comply with a norm) and heterogeneity (a norm describes interactions at a high level independent of implementation details). Researchers have studied norm emergence through social learning where the agents interact repeatedly in a graph structure.

In contrast, we consider norm emergence in an open system, where membership can change, and where no predetermined graph structure exists. We propose Silk, a mechanism wherein a generator monitors interactions among member agents and recommends norms to help resolve conflicts. Each member decides on whether to accept or reject a recommended norm. Upon exiting the system, a member passes its experience along to incoming members of the same type. Thus, members develop norms in a hybrid manner to resolve conflicts.

We evaluate Silk via simulation in the traffic domain. Our results show that social norms promoting conflict resolution emerge in both moderate and selfish societies via our hybrid mechanism.

1 Introduction

A norm characterizes sound or "normal" interactions among the participants of a social group, reflecting their mutual expectations [Singh, 2013]. Norms are a powerful means for regulating interactions among autonomous agents. Familiar examples are driving on the right side of the road in the US and on the left in the UK.

A key application of computational norms is resource sharing, such as information sharing on social media or road sharing by autonomous vehicles, and cybersecurity broadly. Specifically, there is increasing realization of norms for characterizing good behavior and helping achieve secure collaboration [2015]. In essence, norms characterize a social architecture [Singh, 2015] that promotes prosocial behavior.

The goal of this paper is to investigate how norms that resolve conflicts, facilitate coordination, and improve efficiency can emerge, specifically, in a setting of an open sociotechnical system via a hybrid mechanism.

Open. By *open*, we mean that the membership in the system changes dynamically. Existing studies of norm emergence through social learning, e.g., [Airiau *et al.*, 2014], assume the agents interact repeatedly with neighbors in a specified graph. In contrast, we consider a setting wherein outgoing members share their experiences with incoming members. An assumption of a closed system is rarely appropriate. Even a mundane setting, such as cybersecurity within an organization, presupposes openness because the parties concerned are autonomous and changing. Accordingly, we propose a mechanism for norm emergence dubbed *Silk* based on the metaphor of the famous Silk Road—agents enter, transact with each other, and exit. Despite the constant influx of new parties, institutions such as the Silk Road develop norms to regulate their members.

Hybrid, persistent. Previous approaches are either bottom-up (agents come up with the norms) or top-down (a governor determines the norms). We posit that a hybrid approach can provide an effective way to regulate an open MAS.

Silk involves two kinds of agents. A unique *generator* with a global view monitors interactions among two or more *members*. The generator institutes *laws*, i.e., hard integrity constraints that ensure certain undesirable events do not occur. It recommends *norms*, i.e., soft recommendations that if respected would lead to conflict avoidance. A *member* monitors (part of) its environment and takes actions. It is subject to the applicable laws but it can decide whether to adopt any recommended norm. A member receives positive or negative payoffs (possibly different for each member based on its private preferences) from the environment and applies reinforcement learning to update its decision-making policy. Upon exiting, the member shares its experience with incoming members of the same *type*, i.e., those having the same goal as it does.

The generator corresponds to a decision context such as an institution (an enterprise or a marketplace) or a locale (a road intersection or a building). Silk uses a hybrid approach: it combines a top-down view from the generator with bottom-up decision making by the members. This hybrid approach helps overcome three disadvantages of existing studies. First, it provides an overall view of the interactions that individual members lack. Second, it can help improve convergence by recommending norms and thus focusing attention on them. Third, it can improve quality criteria such as fairness.

Sociotechnical. We understand *sociotechnical systems*

(STSs) as systems composed of social agents and technical components [Singh, 2013]. *Integrity* refers to a hard requirement in the domain of interest (e.g., collision in traffic), which an STS cannot allow members to violate. *Conflict* refers to a potential undesirable situation (e.g., an unsafe move in traffic), which an STS allows its members to proceed with (even though it is undesirable) so as to support greater autonomy. Following Jones and Sergot [1993], we differentiate *regimentation* (technical control to ensure integrity that members cannot override) from *regulation* (social control to promote cooperation that members can disregard). STSs need both regimentation and regulation to function effectively.

Silk accommodates both regimentation and regulation. It does so via *laws* and *norms*, respectively. The generator anticipates a potential integrity violation, defines a law to avert it, and broadcasts the law to the members as something they must follow. Think of a law as being realized in the technical infrastructure of an STS. For example, a locking mechanism implements a law prohibiting concurrent access to a resource. The generator also anticipates conflicts, defines a potential norm that would avert the conflict, and broadcasts the norm to the members as a recommendation. Think of a norm as being realized in the social architecture. For example, a norm would be for users to avoid a resource when others are working on it. A recommendation from the generator is not quite a "norm"—a norm based on different members accepting or rejecting the recommendation may come into being.

Contribution. Our contribution is a novel hybrid bottom-up top-down mechanism for norm emergence that supports open MASs and provides constructs to tackle regimentation and regulation in STSs. We find that Silk leads to norm emergence in diverse societies, yields improved performance outcomes, and can enhance fairness in resource sharing.

Organization. Section 2 details Silk and the learning framework. Section 3 describes our traffic intersection scenario. Section 4 offers an empirical evaluation of Silk. Section 5 presents the relevant literature. Section 6 concludes with a discussion of some future research directions.

2 The Silk Framework

Figure 1 shows Silk's architecture: we explain its main components along with the associated algorithms below.

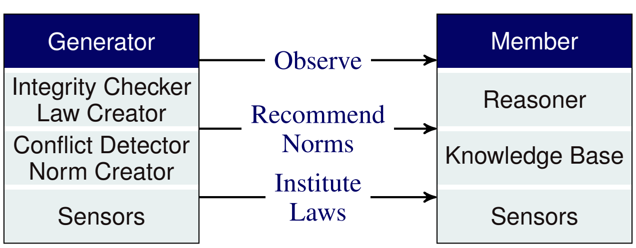


Figure 1: The Silk architecture in schematic terms.

Algorithm 1 shows how the generator generates norms and laws. Its required inputs are: (1) simulation time period T; (2) a list of *views* $\left[\langle v_t^0 \rangle, ..., \langle v_t^m \rangle\right]$ where v_t^i is the i-th view of the particular system state at time t; (3) a grammar G; (4) a function f_{conflict} to detect potential conflicts; and (5) a function $f_{\text{violation}}$ to detect potential integrity violations.

Here, a view is a projection of the system state at a given time. For example, in a traffic scenario a possible view is the traffic state in one part of the road at time t. We express views using predicates describing the system state. Each member has a limited view and can see parts of its environment whereas the generator can see the entire environment.

We adopt grammar G from García-Camino et al. [2009]. This grammar expresses norms and laws in the form $\langle \phi, \Theta(ac) \rangle$, where ϕ is the antecedent of the norm or the law, Θ is a deontic operator, and ac is an action. Examples of norms and laws are provided later.

For each view v_t^i, the generator predicts the next view and detects whether a conflict or an integrity violation would occur based on current laws and recommended norms. For each detected conflict and integrity violation, it generates new norms and laws.

In Algorithm 1, the function $f_{\text{violation}}$ (Line 2) uses the current view (v_t^i) to predict if an integrity violation would occur in the next timestep. If so, the *createL* operator (Line 5) generates a law to prevent that integrity violation. Likewise, the function f_{conflict} (Line 6) uses the current view (v_t^i) to predict if a conflict may occur in the next timestep. If so, the *createN* operator (Line 9) generates a norm.

Both createL and createN place the current view (i.e., v_t^i) as the antecedent of the new norm or law and employ an appropriate deontic operator for the consequent. Specifically, a recommended norm uses permission (per), which indicates neutrality, whereas a law uses prohibition (prh), which indicates a hard restriction. A member's policy (i.e., mapping of states to actions) may end up adopting or rejecting a recommendation, thereby treating the recommended norm as if it were an obligation or prohibition, respectively.

Norm and law generation follows the assumption that if a law or norm would avoid an integrity violation or conflict in the present state, then it would do so in a similar future state.

Algorithm 1 Norm and law generation in Silk

1: **function** Π(views, G, f_{conflict}, $f_{\text{violation}}$, T)
2: violations $\leftarrow$ violationDetection(v_t^i, $f_{\text{violation}}$);
3: **for all** v_t^i in violations **do**
4: **if** there is no law to prevent the violation **then**
5: law $\leftarrow$ createL(G, v_t^i);
6: conflicts $\leftarrow$ conflictDetection(v_t^i, f_{conflict});
7: **for all** members in conflicts **do**
8: **if** there is no norm to prevent the conflict **then**
9: norm $\leftarrow$ createN(G, v_t^i);

Each member has a private payoff matrix and a learning algorithm to reason about its actions. Members apply reinforcement learning to learn behaviors based on their interactions with other members. A member knows neither the identity of the members it interacts with nor their payoff matrices, but can observe their actions. Members use an ϵ-greedy strategy in order to explore the state and action environment. That is, a random action is selected with probability (ϵ) and the action with the highest utility is chosen with probability (1-ϵ).

The exponential function (e^{-Em}) is used to estimate ϵ, where E is a constant and m is the number of times members of the same type experience the same situation. Under this approach, the members mainly explore initially, as they have no prior knowledge to exploit, and gradually increase their extent of exploitation.

Each type of member has two utilities for a norm, for its violation and fulfillment, respectively. Let $U(n,t,a)$ and $U(n,t-1,a)$ be a member's utilities of norm n for action a at times t and $t-1$, $0 \leq \alpha \leq 1$ the learning rate, and $r(n,t,a)$ a reward from the payoff matrix. The utilities are updated as:

$$U(n,t,a) = (1-\alpha) \times U(n,t-1,a) + \alpha \times r(n,t,a). \quad (1)$$

Algorithm 2 shows how the members interact. The generator broadcasts laws and norms. Each member reasons whether to violate or respect each recommended norm and selects an action (Line 3). The members' payoff depends upon their joint actions. They pass on their experiences, represented as average utilities associated with states (antecedent of norms) and actions (violation or fulfillment of norms), to incoming members of the same type (Line 6). In other words, an incoming member can obtain experience from any member who has experience to share: those are members who have been through the current intersection. Sharing experience is thus similar to the situation in human society where old people pass on their experience to young people.

Algorithm 2 Interaction protocol

```
1: generator announces laws and norms;
2: for all members in Conflict do
3:     member selects and performs an action;
4:     compute joint action rewards from payoff matrices;
5:     member updates the utility;
6:     member shares its experience with new members of
       the same type;
```

3 Silk in Traffic

For concreteness and simplicity of exposition, we evaluate Silk on an intersection understood as an open MAS where members continually arrive and depart. The *intersection manager (IM)* is the generator and each *car* is a member.

As Figure 2 shows, we map an intersection and its vicinity to a grid of cells. Traffic flows in each of the four directions. The *(intersection) zone* in the middle is composed of four cells. Cars travel along the grid at the speed of one cell per timestep. Cars can randomly turn left or right at an intersection.

A conflict arises when cars moving in orthogonal directions are about to enter the intersection zone at the same time. These conflicting cars can observe each other's actions but not each other's internal policies. An integrity violation corresponds to a collision: when two cars occupy the same cell. We consider the possibility of collision only within the zone.

Imagine four cars are about to enter the zone, one from each direction. If each proceeds they would occupy distinct cells: hence the IM predicts no integrity violation and produces no law. But it anticipates a conflict (being in the zone at the same time, which we understand as unsafe here), creates a norm, and recommends it to the cars. Each car decides whether to adopt or reject the recommendation. If all reject it, a conflict would occur. Conversely, if all accept it, no progress would occur. We interpret Go as the violation of a recommended norm generated by the IM and Stop as the fulfillment of the generated recommended norms by the IM.

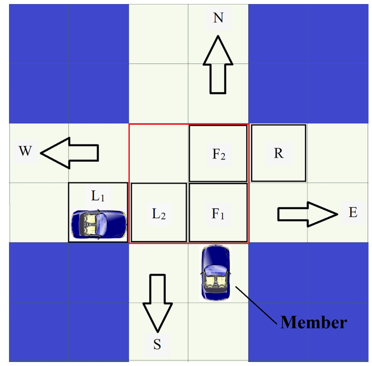


Figure 2: An intersection and (intersection) zone.

Table 1: East-west car's payoffs given two conflicting cars.

	$L_{\text{Go}}, R_{\text{Go}}$	$L_{\text{Stop}}, R_{\text{Go}}$	$L_{\text{Go}}, R_{\text{Stop}}$	$L_{\text{Stop}}, R_{\text{Stop}}$
Go	-5.0	-2.0	-2.0	0.5
Stop	4.0	1.0	1.0	0.0

The antecedent of a norm or a law refers to the values of five cells identified in Figure 2 with respect to the car entering from below: L_1, L_2 to its left, F_1, F_2 to its front, and R to its right. These cells constitute the local view of that car. Our grammar can specify cells with one of six values: $>$ (car heading east), $<$ (car heading west), $\vee$ (car heading south), $\wedge$ (car heading north), $-$ (nothing), and $*$ (car in any heading).

An example norm is $L_1(>)$, $\mathtt{per}$(Go): if a car perceives another car in cell L_1 heading east ($>$), then it is permitted to go (since $\mathtt{per}$ is a neutral deontic, the car also has the option to stop). Figure 2 shows a relevant case where L_1 is occupied but other cells are empty. An example of a law is $L_2(>)$, $\mathtt{prh}$(Go): if a car perceives another car in cell L_2 heading east ($>$), then it is prohibited to go. Tables 1 and 2 show the payoff matrices used in our simulation. These matrices characterize a *moderate* society, where there are higher rewards (1) for north- and southbound cars to Go than to Stop, and (2) for eastbound and westbound cars to Stop than to Go.

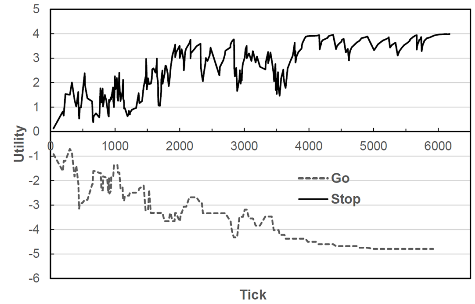
Table 1 shows the payoff matrix for an east- or westbound car that perceives two cars on its left (L) and right (R), traveling north or south. Table 2 gives the payoffs for a north- or southbound car with eastbound or westbound cars on its left and right. When a car conflicts with only one other car, its payoffs are projections of the matrices in Tables 1 and 2 to the first and fourth columns where both Go or both Stop.

Table 2: North-south car's payoffs given two conflicting cars.

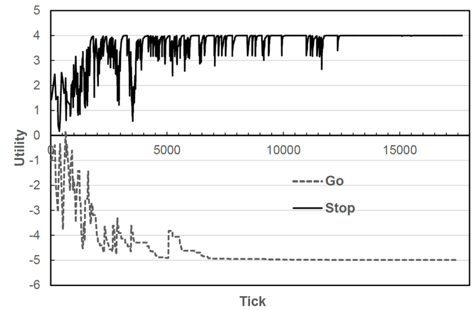
	$L_{\text{Go}}, R_{\text{Go}}$	$L_{\text{Stop}}, R_{\text{Go}}$	$L_{\text{Go}}, R_{\text{Stop}}$	$L_{\text{Stop}}, R_{\text{Stop}}$
Go	-5.0	-2.0	-2.0	4.0
Stop	0.5	0.5	0.5	0.0

Table 3: Generated norms and laws.

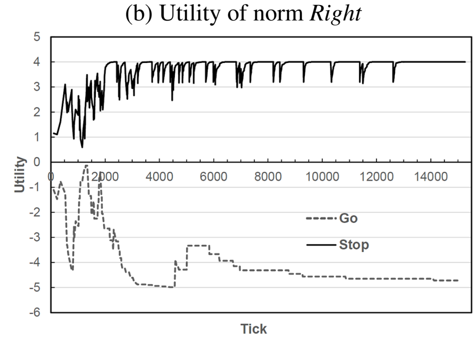
	Antecedent	**Modality**
Left-Right	$L_1(>) \wedge L_2(*) \wedge F_1(*) \wedge F_2(*) \wedge R(<)$	per(Go)
Right	$L_1(-) \wedge L_2(*) \wedge F_1(*) \wedge F_2(*) \wedge R(<)$	per(Go)
Left	$L_1(>) \wedge L_2(*) \wedge F_1(*) \wedge F_2(*) \wedge R(-)$	per(Go)
Collision	$L_1(*) \wedge L_2(>) \wedge F_1(*) \wedge F_2(*) \wedge R(*)$	prh(Go)



(a) Utility of norm *Left-right*



(b) Utility of norm *Right*



(c) Utility of norm *Left*

Figure 3: Utilities of various norms for eastbound cars.

4 Experimental Results

Our simulated traffic system is a simulation environment in Repast [Morales *et al.*, 2013; North *et al.*, 2013]. We consider an intersection where each lane has 20 cells: a total of 76 cells with four cells in the zone. We initialize utilities to zero at $t = 0$ and set α in formula (1) to 0.2. We set $E = 0.05$ in the exponential function (e^{-Em}), which we use for the ϵ-greedy approach. The data shows average over 1,000 trials.

4.1 Norm Emergence

Table 3 shows the norms (first three rows) and law (last row) that IM generates. Left-Right, Right, Left refer to the kind of conflicts they address. Laws come as prohibitions and norm recommendations as permissions. To verify whether conflict-resolving norms can emerge in Silk, we compute the change in average utility for the society from these norms.

According to their directions of travel, the cars converge to a policy on whether to adopt or reject a norm recommendation. As Figure 3 shows, eastbound cars gain utility by stopping in the case of a conflict (whether left, right, or both). All these members acquired the corresponding policy by $t = 12,500$. Figure 4 shows similar results for northbound cars. (The wobbles in utility are due to random selection of actions according to the ϵ-greedy strategy). For brevity, we omit figures for west- and southbound cars: they have the same outcomes as east- and northbound cars, respectively.

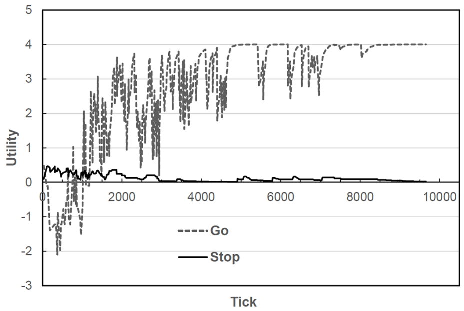
Table 4 summarizes the converged behaviors. Each row describes the matching behavior of cars heading in the specified directions. For example, Row 1 describes what is learned in the conflict situation of Figure 2. The learned behavior is for the eastbound car to yield (Figure 3b) and the northbound car to go (Figure 4c). That is, referring to Table 3, eastbound cars learn to accept IM's recommendation, Right, whereas northbound cars learn to reject IM's recommendation, Left.

4.2 Fairness in Silk

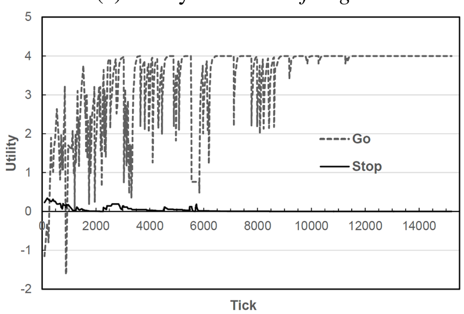
In Table 4, the system converges to the behavior in which the east-west cars stop and north-south cars go. This may not be *fair* [Centeno *et al.*, 2013] to the east-west cars: in heavy north-south traffic, they may be forced into long waits. Therefore, we propose that the generator monitors performance and intervenes if necessary to improve fairness. For example, in our traffic scenario, the IM would monitor delays suffered by the cars and act accordingly. If some cars experience high delays (e.g., five ticks), indicating that they have learned the behavior to stop in a conflict situation, the IM would intervene to let those cars pass the zone by temporarily stopping their conflicting cars from entering the zone. The IM can rely upon the law mechanism to carry this out. We achieve this effect by reversing the converged behavior for one tick.

4.3 Social Performance Improvement in Silk

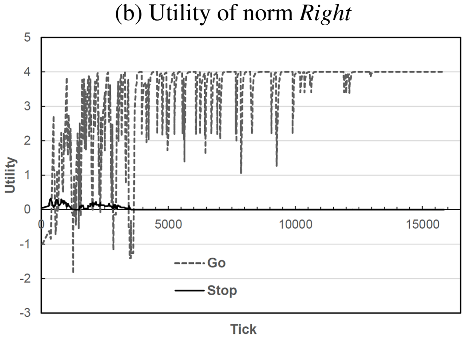
Now we compare Silk with a fully actuated control strategy [FHWA, 2013] as an alternative way of managing traffic. In a fully actuated controller, all approaches have detectors. Each *phase* (i.e., combination of nonconflicting movements) has a *minimum green*, which can be extended if either of the detectors at the approaches detects a car at the stopbar. If no actuation occurs for a given period of time (i.e., *gap time*), the light changes. This change is called a *gap out.* Also, each phase has a *maximum green*, which represents the maximum



(a) Utility of norm *Left-right*



(b) Utility of norm *Right*



(c) Utility of norm *Left*

Figure 4: Utility of norms for northbound cars.

amount of time that a green signal indication can be displayed in the presence of conflicting demand. Maximum green is used to limit the delay to any other movement at the intersection. It corresponds to the fairness property of Silk. In the implemented fully actuated strategy, north- and southbound approaches are paired to form the first phase, and east- and westbound approaches are paired to form the second phase.

To make the fully actuated control strategy as similar as possible to Silk, minimum green is set to one tick and the gap time is set to zero. When a gap out occurs for one phase, which makes the corresponding signals change from green to red, the signals for the other phase can turn green if there is demand (i.e., cars are detected). However, the next phase cannot be implemented until the zone becomes empty. Maximum green is considered five ticks. The described fully actuated logic corresponds to the learned behavior as in Table 4. Since cars can change direction and turn left, in order to prevent collision, we incorporate the law of Table 3 in the logic of the fully actuated controller. We compare Silk with and without the fairness property with traffic signals as well as with and

Table 4: Converged behaviors for cars in different directions.

#	East	West	North	South
1	Yield to right		Go	
2	Yield to left			Go
3		Yield to left	Go	
4		Yield to right		Go
5	Yield to right	Yield to left	Go	
6	Yield to left	Yield to right		Go
7	Yield to both		Go	Go
8		Yield to both	Go	Go
9	Yield to both	Yield to both	Go	Go

Table 5: Silk versus fully actuated control.

	Unfair/No max green		*Fair/Max green*	
	Expected	Average	Expected	Average
Silk	20	25.70	20	24.65
Fully actuated	20	27.75	20	29.41
Improvement		7.40%		16.20%

without maximum green. Our metric is average travel time, calculated by adding expected travel time and the average delay, where delay is the total number of stops for all cars in two successive ticks. Since both Silk and traffic lights are effective in preventing collisions, so it makes sense to compare them in terms of average travel time.

Table 5 shows the average travel time for Silk and the fully actuated controller. As shown, the improvement on travel time gained by using Silk is 7.4% without the fairness property and 16.2% with fairness property.

4.4 Selfish Society

We now investigate the effects of payoff matrices on the converged behaviors. To this end, and to compare with the moderate society we studied above, we study a *selfish* society, as characterized by Table 6. The payoff matrix shows that a member profits from selfishness unless everyone chooses the selfish alternative, in which case everyone loses. Table 6 represents the payoffs for an east-west or north-south car when it perceives two cars on its left and right. When a car conflicts with only one other car, its payoffs are projections of the above matrix to the first and fourth columns where cars from the other two orthogonal directions both go or both stop.

We ran our simulation as before except that we use the above payoff matrices. About half the time (502 times out of 1,000 simulation runs), the north-south cars learn to go and the east-west cars learn to stop (i.e., yield), and in the other

Table 6: East-west and north-south car's payoffs given two conflicting cars.

	$L_{\text{Go}}, R_{\text{Go}}$	$L_{\text{Stop}}, R_{\text{Go}}$	$L_{\text{Go}}, R_{\text{Stop}}$	$L_{\text{Stop}}, R_{\text{Stop}}$
Go	-5.0	-2.0	-2.0	4.0
Stop	0.0	0.0	0.0	0.0

half of the time (498 times out of 1,000 runs of simulation), the reverse behavior emerges. The population converges to one or the other behavior depending upon whether Go or Stop is initially more often selected by east-west or north-south cars. This outcome shows that Silk promotes norm emergence even under radically different payoffs.

4.5 Influence of Disruptive Members

We assumed above that an incoming member uses the experience of old members. Now we study the influence of disruptive members who disregard past experience. We adopt the selfish-society payoff matrices defined in Section 4.4. We assume 10% of the population of east- and westbound members are disruptive: they select Go all the time in conflicting situations. We find that in each of 1,000 simulation runs, the entire population converges to the behavior where east-west cars learn to go and the north-south cars learn to stop.

This experiment brings out the following interesting points. First, a designer can potentially manipulate norms by introducing specific *influencer members*. Second, a hybrid mechanism offers an opportunity to protect against disruptive members by detecting these behaviors and instituting additional laws, possibly by converting existing norms into laws.

5 Related Work

Norms are used to regulate agent behavior and facilitate collaboration in open MASs [Nardin *et al.*, 2016; Xenitidou and Edmonds, 2014], sometimes enforced by punishment that involves both material incentives and normative information [Villatoro *et al.*, 2014]. Savarimuthu and Cranefield [2011] identify five developmental phases of the norm life-cycle: creation, identification, spreading, enforcement, and emergence—and discuss mechanisms for each phase. We focus on the norm emergence phase: we consider a norm as emerged when it is sufficiently widely adopted in the society.

In social learning frameworks [Airiau *et al.*, 2014; Villatoro *et al.*, 2011], agents learn their policies through repeated games with multiple other agents. Agents are considered as nodes of a graph; in each interaction, randomly selected agents play a two-player game. After sufficiently many interactions, social norms evolve in a bottom-up manner. Sugawara [2011] studies a version of the Narrow Road Game using reinforcement learning where two agents traveling in opposite directions decide whether to proceed through a narrow road that has space for at most one agent. Sugawara shows that social norms emerge, but their stability depends upon the agents' characteristics—norms where most agents are selfish are relatively robust. The above papers involve a fixed set of agents, with two agents interacting at a time. In contrast, in Silk varying numbers of agents come, act, and exit (transferring their experience to incoming agents) in a fluid manner.

Morales et al. [2013] proposed a mechanism called IRON for the on-line synthesis of norms. IRON employs designated agents who construct a norm-governed system or institution, and observes the interactions of the members of the system in order to synthesize conflict-free norms in a top-down manner while trying to avoid over-regulation. In subsequent paper [Morales *et al.*, 2015], they proposed an approach, LION, to maximize agents' freedom while maintaining the compactness of the generated norms. LION achieves this by identifying substitutable and complementary norms and using them to synthesize liberal normative systems at run-time. Silk is fundamentally different from Morales et al.'s approaches. Whereas their approaches adopt a central norm learner, Silk employs a hybrid approach, as explained, and tackles the problem of fairness.

Most previous normative MAS approaches consider norms as behaviors that may become more strict, i.e., acquire sanctions for violation [Savarimuthu and Cranefield, 2011] or treat a norm as just an element of a law [Paes *et al.*, 2005]. However, some systems handle hard requirements (integrity in our terms) via laws and soft requirements (conflict avoidance in our terms) via norms. Computing approaches outside of MAS, e.g., [Minsky and Ungureanu, 2000], tackle engineering regimented systems but do not incorporate norms. Silk is the first work to handle laws and norms distinctly and apply them both to open systems.

Agent-based technologies have been applied to the field of traffic and transportation in recent years [Bazzan and Klügl, 2014], including auctions [Carlino *et al.*, 2013; Mashayekhi and List, 2015], and intersection management [Au *et al.*, 2015], and learning [Tumer *et al.*, 2009]. Researchers have investigated how to minimize total travel time or queue length [Vasirani and Ossowski, 2009]. Few papers tackle the intersection management problem from the viewpoint of norms. Our application of Silk in the intersection setting fills this gap.

6 Conclusions and Future Work

Silk is a hybrid bottom-up top-down mechanism for norm emergence. Silk is expressly suited to systems that are open (changing membership and fluid interactions determined on the fly). It accommodates regimentation and regulation in sociotechnical systems: importantly, the existence of laws promotes the emergence of norms across a variety of societies, ranging from moderate to selfish. Silk yields improved outcomes over a traditional, fully actuated approach both in settings where fairness is ensured and where it is not.

Silk can be potentially applied in any open MAS that satisfies these criteria: (1) its members compete over resources, which may lead to conflicts and integrity violations, (2) at least one agent can observe all the members, (3) members can observe their neighbors and can communicate. An important application is in the regulation of self-driving cars controlled by agents that communicate with each other or the infrastructure and learn what actions to take, thus relieving humans from driving and making decisions when approaching shared spaces, such as intersections.

One important direction is implementing Silk for information domains in which cybersecurity and privacy are crucial. Norms in settings such as online communities concern information sharing (e.g., is sharing pictures of gunshot victims and perpetrators appropriate?), which could involve sanctions via complaints and direct criticisms.

Acknowledgments

We thank the US Department of Defense for support through the Science of Security Lablet grant to NC State University.

References

[Airiau *et al.*, 2014] Stéphane Airiau, Sandip Sen, and Daniel Villatoro. Emergence of conventions through social learning. *Autonomous Agents and Multi-Agent Systems*, 28(5):779–804, 2014.

[Au *et al.*, 2015] Tsz-Chiu Au, Shun Zhang, and Peter Stone. Autonomous intersection management for semi-autonomous vehicles. In Dusǎn Teodorovíc, editor, *The Routledge Handbook of Transportation*, chapter 7, pp. 88–104. Taylor & Francis, 2015.

[Bazzan and Klügl, 2014] Ana L. C. Bazzan and Franziska Klügl. A review on agent-based technology for traffic and transportation. *The Knowledge Engineering Review*, 29:375–403, 2014.

[Carlino *et al.*, 2013] D. Carlino, S. D. Boyles, and P. Stone. Auction-based autonomous intersection management. *Proc. International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 529–534, 2013.

[Centeno *et al.*, 2013] Roberto Centeno, Holger Billhardt, and Ramón Hermoso. Persuading agents to act in the right way: An incentive-based approach. *Engineering Applications of Artificial Intelligence*, 26(1):198–210, 2013.

[Du *et al.*, 2015] Hongying Du, Bennett Y. Narron, Nirav Ajmeri, Emily Berglund, Jon Doyle, and Munindar P. Singh. Understanding sanction under variable observability in a secure, collaborative environment. *Proc. International Symposium and Bootcamp on the Science of Security (HotSoS)*, pp. 12:1–12:10, ACM, 2015.

[FHWA, 2013] FHWA. Traffic signal timing manual, 2013. Federal Highway Administration.

[García-Camino *et al.*, 2009] Andrés García-Camino, J. A. Rodríguez-Aguilar, Carles Sierra, and Wamberto Vasconcelos. Constraint rule-based programming of norms for electronic institutions. *Autonomous Agents and Multi-Agent Systems*, 18(1):186–217, 2009.

[Jones and Sergot, 1993] Andrew J. I. Jones and Marek Sergot. On the characterisation of law and computer systems. *Deontic Logic in Computer Science*, pp. 275–307, 1993.

[Mashayekhi and List, 2015] Mehdi Mashayekhi and George List. A multiagent auction-based approach for modeling of signalized intersections. *IJCAI Workshops on Synergies Between Multiagent Systems, Machine Learning and Complex Systems*, pp. 13–24, 2015.

[Minsky and Ungureanu, 2000] Naftaly H. Minsky and Victoria Ungureanu. Law-governed interaction: A coordination and control mechanism for heterogeneous distributed systems. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 9(3):273–305, 2000.

[Morales *et al.*, 2013] Javier Morales, Maite López-Sánchez, J. A. Rodríguez-Aguilar, Michael Wooldridge, and Wamberto Vasconcelos. Automated synthesis of normative systems. *Proc. AAMAS*, pp. 483–490, 2013.

[Morales *et al.*, 2015] Javier Morales, Maite López-Sánchez, J. A. Rodríguez-Aguilar, Michael Wooldridge, and Wamberto Vasconcelos. Synthesising liberal normative systems. *Proc. AAMAS*, pp. 433–441, 2015.

[Nardin *et al.*, 2016] Luis G. Nardin, Tina Balke-Visser, Nirav Ajmeri, Anup K. Kalia, Jaime S. Sichman, and Munindar P. Singh. Classifying sanctions and modelling a sanctioning process for socio-technical systems. *The Knowledge Engineering Review*, 31(2):142–166, 2016.

[North *et al.*, 2013] Michael J. North, Nicholson T. Collier, Jonathan Ozik, Eric R. Tatara, Charles M. Macal, Mark Bragen, and Pam Sydelko. Complex adaptive systems modeling with Repast Simphony. *Complex Adaptive Systems Modeling*, 1(1):1–26, 2013.

[Paes *et al.*, 2005] Rodrigo Paes, G. R. de Carvalho, C. J. P. de de Lucena, P. S. C. Alencar, H. O. de Almeida, and V. T. da Silva. Specifying laws in open multi-agent systems. *Proc. Workshop on Agents, Norms and Institutions for Regulated Multi-agent Systems (ANIREM)*, 2005.

[Savarimuthu and Cranefield, 2011] Bastin Tony Roy Savarimuthu and Stephen Cranefield. Norm creation, spreading and emergence. *Multiagent and Grid Systems*, 7(1):21–54, 2011.

[Singh, 2013] Munindar P. Singh. Norms as a basis for governing sociotechnical systems. *ACM Transactions on Intelligent Systems and Technology*, 5(1):21:1–21:23, 2013.

[Singh, 2015] Munindar P. Singh. Cybersecurity as an application domain for multiagent systems. *Proc. AAMAS*, pp. 1207–1212, 2015.

[Sugawara, 2011] Toshiharu Sugawara. Emergence and stability of social conventions in conflict situations. *Proc. IJCAI*, pp. 371–378, 2011.

[Tumer *et al.*, 2009] K. Tumer, A. Agogino, and Z. Welch. Traffic congestion management as a learning agent coordination problem. In A. Bazzan and F. Kluegl, editors, *Multiagent Architectures for Traffic and Transportation Engineering*, pp. 261–279. IGI Global, 2009.

[Vasirani and Ossowski, 2009] Matteo Vasirani and Sascha Ossowski. A market-inspired approach to reservation-based urban road traffic management. *Proc. AAMAS*, pp. 617–624, 2009.

[Villatoro *et al.*, 2011] Daniel Villatoro, Jordi Sabater-Mir, and Sandip Sen. Social instruments for robust convention emergence. *Proc. IJCAI*, pp. 420–425, 2011.

[Villatoro *et al.*, 2014] Daniel Villatoro, Giulia Andrighetto, Jordi Brandts, Luis Gustavo Nardin, Jordi Sabater-Mir, and Rosaria Conte. The norm-signaling effects of group punishment combining agent-based simulation and laboratory experiments. *Social Science Computer Review*, 32(3):334–353, 2014.

[Xenitidou and Edmonds, 2014] Maria Xenitidou and Bruce Edmonds. *The Complexity of Social Norms*. Springer, 2014.